ABSTRACT

The specific purpose of this thesis is the automated recognition of the off-line Chinese hand-printed characters by using a blue ball-point pen. Through mask processing, the main components in a Chinese character such as vertical, horizontal, and slant strokes can be extracted. Then, the connected components with the coordinates of the top, bottom, leftmost, and rightmost ends of each stroke extracted are found. From these coordinates, the length and position of each stroke can be computed.

According to the number, relative length, and relative position of each stroke, both of the coarse and fine rule-based classification can be made, and the goal of this thesis is able to be reached.

Excluding the load and segmentation of the original image, the computing time for the feature extraction and classification depends on the image size and the number of strokes. It is about 0.3 seconds per Chinese character on an IBM PC 80486 DX33.

The advantages of the proposed method include efficient time complexity, strong ability to detect very similar Chinese characters, tolerance of the slope of the stroke, and 96% or higher recognition rate.

The disadvantage is the inflexibility for learning driven by the users since the matching rules are open to the manufactures only at present.

OFF-LINE HAND-PRINTED CHINESE CHARACTER RECOGNITION BASED ON STROKE MATCHING

by
Mr. Sunshine Chang

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

Department of Computer and Information Science

January 1995

APPROVAL PAGE

OFF-LINE HAND-PRINTED CHINESE CHARACTER RECOGNITION BASED ON STROKE MATCHING

Sunshine Chang

Dr. Frank Y. Shih, Thesis Advisor — Date
Associate Professor of Department of
Computer and Information Science

Dr. Peter A. Ng, Committee Member — Date
Professor and Chairman
Department of Computer and Information Science

Dr. James McHugh, Committee Member — Date
Graduate Advisor and Ph.D. Program Director
Department of Computer and Information Science

BIOGRAPHICAL SKETCH

Author: Mr. Sunshine Chang

Degree: Master of Science in Computer Science

Date: January 1995

Date of Birth:

Place of Birth:

Undergraduate and Graduate Education

- Master of Science in Computer Science,
  New Jersey Institute of Technology
  Newark, New Jersey, 1995

- Bachelor of Science in Computer Science,
  Tamkang University
  Taiwan, R.O.C., 1990

Major: Computer Science

This thesis is dedicated to
my parents.

ACKNOWLEDGMENT

The author wishes to express his sincere gratitude to his advisor, Dr. Frank Shih, for his guidance, friendship, and moral support throughout this research.

Special thanks to Professors Peter A. Ng and James McHugh for serving as members of the committee.

The author appreciates the timely help and suggestion from his classmates, friends, relatives, and families.

And finally, a thank you to all of the authors and their advisors listed in the references of this thesis for their experiences.

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Optical Characters Recognition enables a computer to recognize the characters written or printed on a paper or certain facility. It is a branch of Pattern Recogniton, whose goal is to make a computer to distinguish among objects. The ambition of this domain is Computer Vision, which makes efforts on how a computer can "see" a 3D object in the real world as we, the humans, do. The basic common principles are how to define an object being recognized, to extract useful features from it, and to match the patterns.

Two of the important and pratical applications of Optical Characters Recognition are for the post office, and for the blind. The others include automatic data entry, document processing, etc.

According to the input methods and facilities, the Opitical Character Recognition, or OCR, can be classified into two classes, on-line ,and off-line. Usually, the characters are input one by one in an on-line OCR, page by page in an off-line OCR.

The objects being recognized in an off-line OCR may be divided to four kinds: machine-printed, hand-printed, hand-written, and script characters. The hand-printed characters are written carefully. There are always much more different writting habits of certain writers existed in hand-written characters than in hand-printed characters. The script characters may join or connect together.

Machine recognition of hand-printed or written Chinese characters is very difficult due to the following factors:

1. High complexity exists in the hand-printed or written characters because of the variation caused by the writing instruments, paper surfaces, social environment, and the writing habits, education background, mood, and helth of the writers.

2. Compared to the total number (26x4) of English characters, the total number of Chinese characters is very huge. It is more than 40,000 totally, and 5,401 commonly used.

3. Almost 10% of Chinese characters such as 鹽, 盪, and 黴 have more than 20 strokes.

4. Some characters are very similar, e.g. 己, 已, 巳, and 日, 曰.

Generally, the methods used in an off-line OCR may have image file processing, binarization, text/graphics/image segmentation, characters segmentation, thinnig, normalization, feature extraction, coares classification, and fine classification.

The proposed methods are different from the above in some degree. Instead of the thinning and normalization, dilation are used. And this thesis focuses mainly on the feature extraction of Chinese characters. Besides, a set of simplified matching rules is created to verify the proposed system that can work in an acceptable recognition rate and speed. However, for the purpose of commercialization, the matching algorithm should be revised and expanded.

CHAPTER 2

LITERATURE SURVEY

Geometrical and topological feature analysis method may represent characters' global and local properties including strokes, bays, end points, intersection of line segments, loops, and stroke relations, angular properties, sharp protrusions. These features have high tolerances to distortion, style variation, and some degree of rotation.

The methods of stroke extraction using traditional image processing have been studied. Chang [6] proposed an accumulated and learnable multifont (printed) OCCR, extracting the strokes by choosing the maximum run length of each object pixel in 4 directions of degrees 0, 45, 90, and 135. But it is not good at distinguishing from the similar characters.

Another Chang [7] proposed a method of strokes extraction of Chinese characters by selecting the largest run length of every object pixel from 8 directions of degrees 0, 22.5, 45, 67.5, 90, 112.5, 135, and 157.5. Its disadvantage is too sensitive. And the time complexty is not good for a non parallel computer.

Chao [8] proposed a new stroke extraction method for hand-written characters based on the adjacency, branching, width, and length of runs of object pixels in horizontal direction only. The result of stroke extraction looks good, however, he did not prove whether or not the feature extrated in his method can be applied on the matching process correctly and efficiently.

Chen[14] proposed a method of recognition of handwritten Chinese characters via short-line segments representing the strokes. The recognition rate is 90.88%, and the speed is 8.73 seconds per character.

The neural network applied on hand-printed or written Chinese characters recognition requires a large scale of neocognitron, and it is time-consuming to design the training patterns for feature extraction. Besides, the recognition rate is not ideal for the pattern which has not been trained. The parallel processing potential is an important characteristic of neural network, however it can not be developed easily on the popular personal computer at present. For illustration, it takes 28 seconds to recognize the hand-printd Chinese character with 86% recognition rate even on an IRIS workstation according to [11]. In [12], the recognition rate is 75% without rejection in the speed of 2 seconds per handwritten Chinese character on an IBM PC 386. In [13], 230 seconds are required even for the recognition of a handwritten capital English character on an IBM PC 386.

Context or language model [9][10] is also introduced into the OCCR. It is expected to help the recognition process to select the correct character from candidate sets produced by the coarse classification. However it can not ensure a satisfied recognition rate in a reasonable cost, time and space complexity, because of the propogated error resulted from:

1. incorrect character in original text,
2. correct character in original text, but
   a. the correct character not included in the candidate sets, which is the result of the coarse classification,
   b. the inadequate decision policy of the language model itself,
   c. the incompleteness of the word base of the language model.

CHAPTER 3

THE PROPOSED METHOD

3.1 Objective

The objective of this thesis is to propose an improved traditional image processing method in which a popular personal computer can recognize off-line a scanned image of Chinese characters printed on a white paper by hand using a common blue ball-point pen.

3.2 Materials And Experimental Methods

Materials related to the experiment of the proposed method include an IBM PC 80486 DX33, a 300 dpi flat-bed grey level scanner, and a printer. The software is implemented in the C language.

The training and testing data are sampled from the same 500 Chinese characters written (hand-printed) by each of 15 persons with different ages and education background. The character size may be from 18x18 to 60x60 pixels. The size of 40x40 pixels is suggested.

The proposed system includes modules of image-file processing, preprocessing, feature extraction, post-processing, and classification.

3.3 Image-file Processing

In this thesis, the image of a Chinese character is scanned into a computer as the form of the grey level image file.

The coordinate system used is column major, and the origin is at the top-left corner.

The Constrained Run Length Algorithm [3][4][5] is used in the segmentation process.

Since so many tools can be found and used, this thesis doesn't focus on the simple process of open, display[2], close, and segmentation of a document image file.

```
         012345
Origin +---------->   Y
      0|
      1|
      2|
      3|
      4|
      5|
       |
       |

       X
```

255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	251	255	253	242	248	211	110	118	243	251	255	250	255	255	255	255	254	255	255	255	255	255	255	255	253	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	254	255	255	185	113	169	250	255	255	254	255	255	255	255	255	255	255	255	255	253	251	247	255	255
255	255	255	255	255	255	255	255	251	255	255	255	255	255	255	255	254	249	249	242	192	116	164	252	255	255	248	252	255	255	253	253	255	255	255	249	253	251	254	255	255
255	255	255	255	255	255	255	255	255	255	255	251	255	255	255	255	255	248	255	244	186	109	178	250	249	246	249	248	248	255	255	253	255	255	255	255	255	255	255	255	255
255	254	255	255	255	255	254	253	255	255	255	255	255	253	255	250	253	251	254	246	165	103	184	243	242	246	251	254	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	253	255	241	240	235	153	114	212	255	255	255	252	255	255	255	255	255	255	255	255	255	255	255	255	255	254
255	255	255	255	255	255	252	255	255	255	255	255	255	255	255	255	255	255	255	248	146	121	228	255	253	255	255	243	255	251	255	247	255	248	255	255	254	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	244	133	108	222	253	249	254	255	230	242	247	250	255	248	255	255	255	251	252	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	251	238	123	110	211	254	253	248	252	247	203	254	248	244	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	120	111	227	255	254	255	255	255	211	187	244	255	255	255	255	255	253	255	255	255	255
255	255	255	253	255	255	255	255	255	255	255	255	255	255	255	255	247	255	254	238	124	112	230	251	255	255	255	255	233	155	213	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	248	247	237	121	117	233	255	255	255	255	255	243	159	141	227	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	253	246	233	115	110	213	255	255	255	252	246	255	225	124	152	251	255	255	255	255	255	255	255	255
255	255	249	252	255	255	253	255	255	255	255	255	255	255	254	255	251	251	254	244	107	100	225	244	244	241	246	247	248	219	131	107	169	253	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	252	255	255	255	252	250	223	101	105	215	248	252	244	231	225	170	113	102	115	185	248	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	248	216	97	106	202	240	240	227	198	128	108	114	125	160	248	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	254	255	249	249	253	250	246	245	248	227	103	101	205	196	144	112	106	112	147	185	214	235	253	255	255	255	255	255	255	255	255
252	249	253	252	253	252	252	255	255	246	247	251	254	250	244	242	245	244	235	204	96	84	104	99	105	104	140	177	233	248	255	255	255	255	255	255	250	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	254	243	230	223	235	212	174	129	97	75	72	93	106	143	188	216	247	250	253	255	254	255	252	255	255	255	255	255	255	255
255	255	255	250	255	254	255	255	255	255	255	255	249	225	176	134	94	88	90	96	77	86	153	215	238	248	255	255	255	255	255	255	255	255	254	255	255	255	255	255	255
253	255	255	255	255	255	254	255	252	246	239	203	137	102	101	97	102	123	138	168	80	90	166	251	230	241	249	255	255	255	255	255	255	255	252	251	255	255	255	255	255
251	255	255	255	250	250	252	250	218	166	114	96	100	103	119	139	169	213	233	211	84	96	152	218	135	167	227	249	255	255	255	254	255	255	250	255	255	255	255	253	255
255	255	255	253	246	246	197	145	96	86	89	97	120	171	221	232	250	238	248	181	85	103	184	226	158	114	116	199	237	241	255	253	255	255	255	255	255	255	255	255	255
255	255	255	240	219	142	88	86	94	116	152	190	234	252	249	255	254	241	219	96	93	106	221	245	219	126	97	114	205	251	255	255	255	255	255	255	255	255	255	255	255
254	253	224	115	122	121	116	115	177	222	248	246	248	254	250	251	252	239	179	101	95	106	226	251	243	205	111	94	99	186	245	246	255	255	255	254	254	255	255	255	255
255	254	149	139	153	163	200	229	249	255	255	255	255	255	251	250	239	218	113	116	114	108	221	253	239	245	183	112	103	98	185	253	255	255	255	255	252	251	254	255	255
255	255	232	208	228	230	248	251	255	255	255	255	255	255	255	240	227	164	109	163	112	104	218	245	248	250	234	190	109	91	93	149	239	248	254	255	250	255	255	255	255
255	255	251	254	255	255	255	255	255	255	255	255	255	244	248	255	223	115	107	204	105	109	219	252	255	254	255	249	209	106	89	84	156	244	254	255	255	255	255	255	255
254	252	255	255	255	255	255	255	255	255	255	255	255	255	255	245	157	107	160	222	105	109	221	255	255	255	255	255	247	201	125	95	87	127	226	250	255	255	255	255	255
253	250	252	255	255	255	255	255	255	255	255	255	255	255	253	197	109	126	215	228	110	108	232	255	255	255	253	251	254	248	232	132	104	84	108	203	247	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	254	250	230	129	114	192	243	233	103	116	230	252	255	255	255	255	255	255	255	230	178	104	83	96	161	236	255	255	255
255	255	255	255	255	255	255	255	255	255	255	252	250	239	151	113	139	239	255	227	109	115	216	248	251	255	255	255	255	255	255	255	255	210	132	100	96	107	129	220	253
255	255	255	255	255	255	255	255	255	255	255	251	248	181	110	128	213	244	255	239	114	111	224	255	255	255	255	255	252	255	255	255	255	254	246	186	122	99	92	103	142
255	255	255	255	255	255	255	255	255	255	255	255	196	118	120	196	249	255	255	237	120	119	224	255	255	255	255	255	255	255	255	255	255	255	255	254	231	204	137	114	101
250	255	255	255	255	255	249	254	255	253	255	219	120	107	157	246	255	244	255	241	132	117	226	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	243	193	121
255	255	255	255	255	255	255	255	255	255	242	142	118	140	238	248	245	253	254	253	148	114	225	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	222
255	255	255	255	254	255	255	255	253	253	201	121	131	209	246	252	255	254	253	255	167	117	204	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
246	255	255	255	255	255	254	255	248	208	128	124	196	243	246	255	255	255	255	252	179	114	189	255	255	255	255	255	255	255	255	254	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	248	222	141	122	168	245	253	254	251	255	255	255	255	170	109	181	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	254	255	253	233	159	122	159	240	254	255	255	255	255	255	255	251	161	107	175	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
249	255	255	248	252	255	246	191	122	132	231	250	253	255	255	255	255	255	255	255	173	117	208	255	255	255	255	255	255	253	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	217	130	114	188	255	255	255	255	255	255	255	255	219	237	158	122	203	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	231	157	116	163	254	255	255	255	255	255	255	255	253	229	235	152	109	202	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	253	241	158	121	138	224	255	255	255	255	255	255	255	255	252	232	240	163	112	217	252	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	241	187	124	140	207	255	255	255	255	255	255	255	255	252	248	233	233	175	118	225	255	254	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	229	139	135	219	255	255	255	255	255	255	255	255	255	255	255	234	244	169	112	222	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	241	142	129	194	252	255	255	255	255	255	255	255	255	252	255	255	194	195	134	131	234	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	248	172	123	166	255	255	255	255	255	255	255	255	255	255	255	255	245	210	186	136	127	230	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
235	159	137	163	233	255	255	255	255	255	255	255	255	255	255	250	255	255	195	176	130	128	243	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
184	145	174	241	255	254	253	250	251	255	255	255	255	255	248	250	249	240	183	139	122	133	243	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
135	172	243	255	255	255	250	251	255	251	255	255	255	255	255	255	253	253	216	144	122	134	248	255	255	255	255	255	255	255	255	252	255	255	255	255	255	255	255	255	255
164	238	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	244	133	109	139	242	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
247	255	255	255	254	255	255	255	255	255	255	255	249	252	255	255	255	255	243	157	121	172	246	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255

Figure 1 Original image of Chinese character 木 represented by grey scale from 0, denoting black, to 255, denoting white.

3.4 Preprocessing

Before the feature extraction is performed, some preprocessing is required in order to delete noise and to improve the precesion of feature extracted. First the thresholding and binarization process is applied.

By thresholding and binarization, the scanned-in grey level image file of a Chinese character might be transformed into a form of a binary image file, which would save more memory space. However, the image-file processing system is preferred to remain the form of grey level, because the level from 1 to 254 can be used to denote the intermediate result of some process on a pixel in this image besides of the level 0 denoting the object and 255 denoting the background.

Then the dilation is executed. Its goal is to make the object at least two pixels thick. The following is the algorithm of the dilation.

Algorithm of Dilation:

```
if (  (DilatedBinarizedOriginal[i-1][j]>1 )
    &&(DilatedBinarizedOriginal[i  ][j]<1 )
    &&(DilatedBinarizedOriginal[i+1][j]>1 ))
       DilatedBinarizedOriginal[i-1][j] = 0;

if (  (DilatedBinarizedOriginal[i][j-1]>1 )
    &&(DilatedBinarizedOriginal[i][j  ]<1 )
    &&(DilatedBinarizedOriginal[i][j+1]>1 ))
       DilatedBinarizedOriginal[i][j-1] = 0;
```

Figure 2 Preprocessing result of figure 1

3.5 Feature Extraction

There are many kinds of features which can be analyzed from Chinese characters. However, the main feature considered in this thesis is stroke including vertical, horizontal, and slant one.

Another feature considered is the number of the three kinds of strokes. The others are their relative position and length.

```
0 0 0    0 0 0   0-1 0   0 0 0
0 1 0   -1 1 0   0 1 0   0 1-1
0-1 0    0 0 0   0 0 0   0 0 0
```

Four masks as above are designed for the detection of upper, right-hand side, lower, and left-hand side boundary separately. By the following two algorithms, the vertical and horizotal strokes can be extracted.

Algorithm of vertical feature extraction:

```
if ((ResultOfMask1[i][j]=1)||(ResultOfMask3[i][j]=1))
{
  VerticalFeature[i][j]=1;/* Hidden as black background */
}
else if(ResultOfPreprocessing[i][j]==255) /* white
                                             background */
{
  VerticalFeature[i][j]=0;  /* black background */
}
else
```

```
{
  VerticalFeature[i][j]=255;/* white object */
}
```

Algorithm of horizontal feature extraction

```
if ((Result2[i][j]=1)||(Result4[i][j]=1))
{
  HorizontalFeature[i][j]=1;/* Hide as black background */
}
else if(Result0[i][j]>=100) /* white background */
{
  HorizontalFeature[i][j]=0;/* black background */
}
else
{
  HorizontalFeature[i][j]=255;/* white object */
}
```

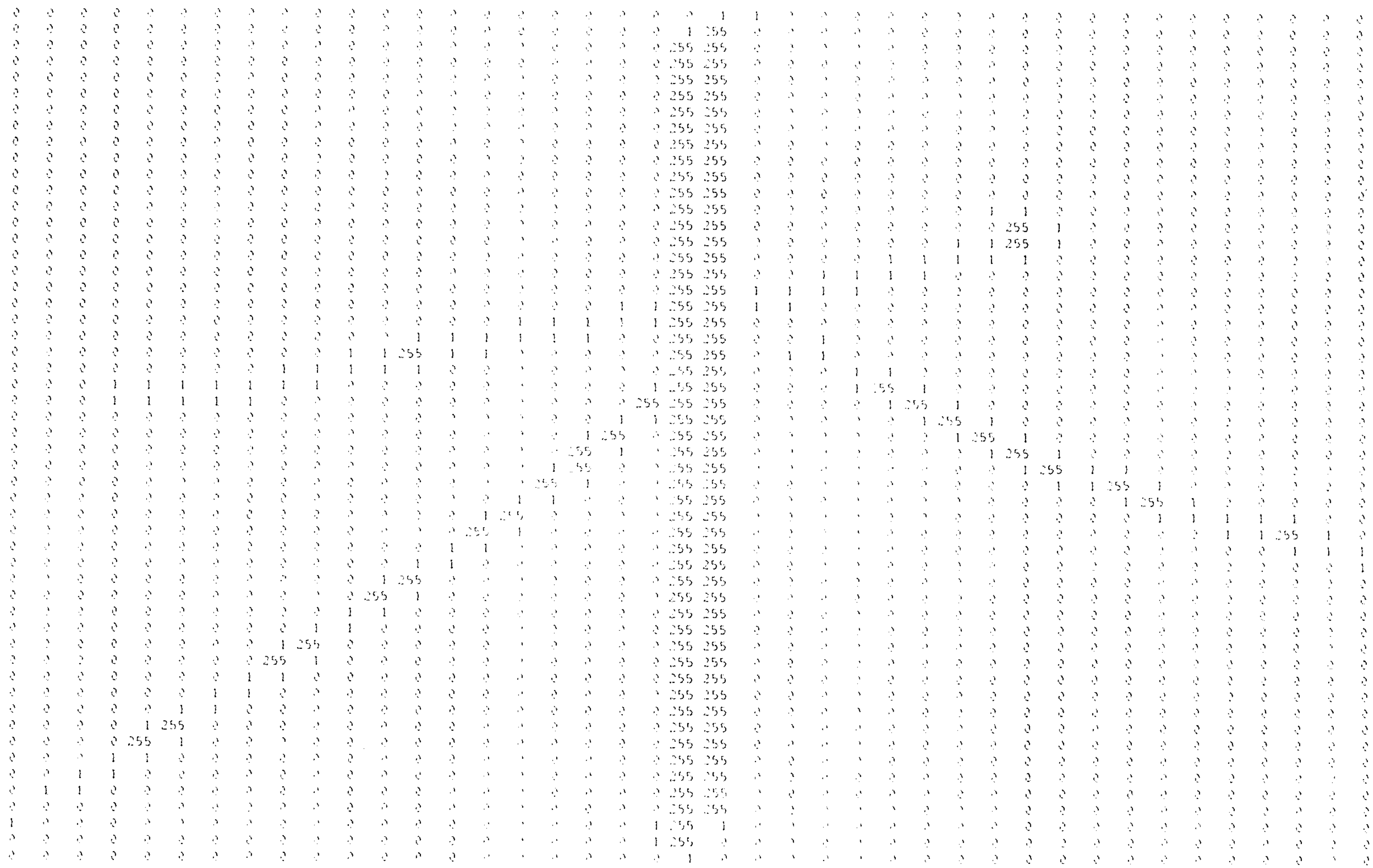

Figure 3 Noisy result of vertical feature extraction

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	12	55	255	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	255	2552	55	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	255	255	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	255	255	255	255	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	255	255	255	255	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	255	255	255	255	255	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	255	255	255	255	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	255	255	255	1	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	255	255	255	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	255	255	255	255	255	1	0	0	0	0	0	0	0	0	0	1	255	1	0	0	0	1	255	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	255	255	255	1	0	0	0	0	0	0	0	0	0	0	0	1	255	1	0	0	0	0	1	255	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	255	255	1	0	0	0	0	0	1	255	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0	12	55	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0	1	255	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	1	255	255	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	1	255	255	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	255	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	255	255	255	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	255	255	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	255	1
0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	255	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4 Noisy result of horizontal feature extraction

The feature extracted previously can not be satisfied. Some more post-processes including the deletion of single points, the breaking of the undesired connection, and the thresholding of the stroke length are required.

Algorithm of deletion of single points:

```
if(  (VerticalFeature[i-1][j-1] <=1 )
   &&(VerticalFeature[i  ][j-1] <=1 )
   &&(VerticalFeature[i+1][j-1] <=1 )
   &&(VerticalFeature[i-1][j  ] <=1 )
   &&(VerticalFeature[i  ][j  ]  >1 )
   &&(VerticalFeature[i+1][j  ] <=1 )
   &&(VerticalFeature[i-1][j+1] <=1 )
   &&(VerticalFeature[i  ][j+1] <=1 )
   &&(VerticalFeature[i+1][j+1] <=1 ))
      VerticalFeature[i][j] = 0;
if(  (HorizontalFeature[i-1][j-1] <=1 )
   &&(HorizontalFeature[i  ][j-1] <=1 )
   &&(HorizontalFeature[i+1][j-1] <=1 )
   &&(HorizontalFeature[i-1][j  ] <=1 )
   &&(HorizontalFeature[i  ][j  ]  >1 )
   &&(HorizontalFeature[i+1][j  ] <=1 )
   &&(HorizontalFeature[i-1][j+1] <=1 )
   &&(HorizontalFeature[i  ][j+1] <=1 )
   &&(HorizontalFeature[i+1][j+1] <=1 ))
      HorizontalFeature[i][j] = 0;
```

Algorithm of breaking the undesired connection:

```
if(  (VerticalFeature[i][j-3] >1 )
   &&(VerticalFeature[i][j-2] >1 )
   &&(VerticalFeature[i][j-1] >1 )
   &&(VerticalFeature[i][j  ] >1 ))
      VerticalFeature[i][j  ] =1;
if(  (HorizontalFeature[i-3][j] >1 )
   &&(HorizontalFeature[i-2][j] >1 )
   &&(HorizontalFeature[i-1][j] >1 )
   &&(HorizontalFeature[i  ][j] >1 ))
      HorizontalFeature[i  ][j] =1;
if(  (VerticalFeature[i-3][j-3] >1 )
   &&(VerticalFeature[i-2][j-2] >1 )
   &&(VerticalFeature[i-1][j-1] >1 )
   &&(VerticalFeature[i  ][j  ] >1 ))
      VerticalFeature[i  ][j  ] =1;
if(  (HorizontalFeature[i-3][j-3] >1 )
   &&(HorizontalFeature[i-2][j-2] >1 )
   &&(HorizontalFeature[i-1][j-1] >1 )
   &&(HorizontalFeature[i  ][j  ] >1 ))
      HorizontalFeature[i  ][j  ] =1;
if(  (VerticalFeature[i+3][j-3] >1 )
   &&(VerticalFeature[i+2][j-2] >1 )
   &&(VerticalFeature[i+1][j-1] >1 )
   &&(VerticalFeature[i  ][j  ] >1 ))
      VerticalFeature[i  ][j  ] =1;
```

```
if(  (HorizontalFeature[i+3][j-3] >1 )
   &&(HorizontalFeature[i+2][j-2] >1 )
   &&(HorizontalFeature[i+1][j-1] >1 )
   &&(HorizontalFeature[i  ][j  ] >1 ))
      HorizontalFeature[i  ][j  ] =1;
```

Then, the connected components and the coordinates of the top, bottom, left-most, and right-most ends of them are found out. From these coordinates, the relative length and position can be easily computed.

If the length is less than a threshold value, then this connected component might be a noise and should be deleted. According to the observation and thousands of experiments, this value depends on each individual character and is very critical to feature extraction, and even to the success of character recognition. However, it is assigned initially a value of one tenth of the average of width and length of the object being recognized. If the processes of extraction, matching, or recognition are not satisfactory, this threshold value could be adjusted.

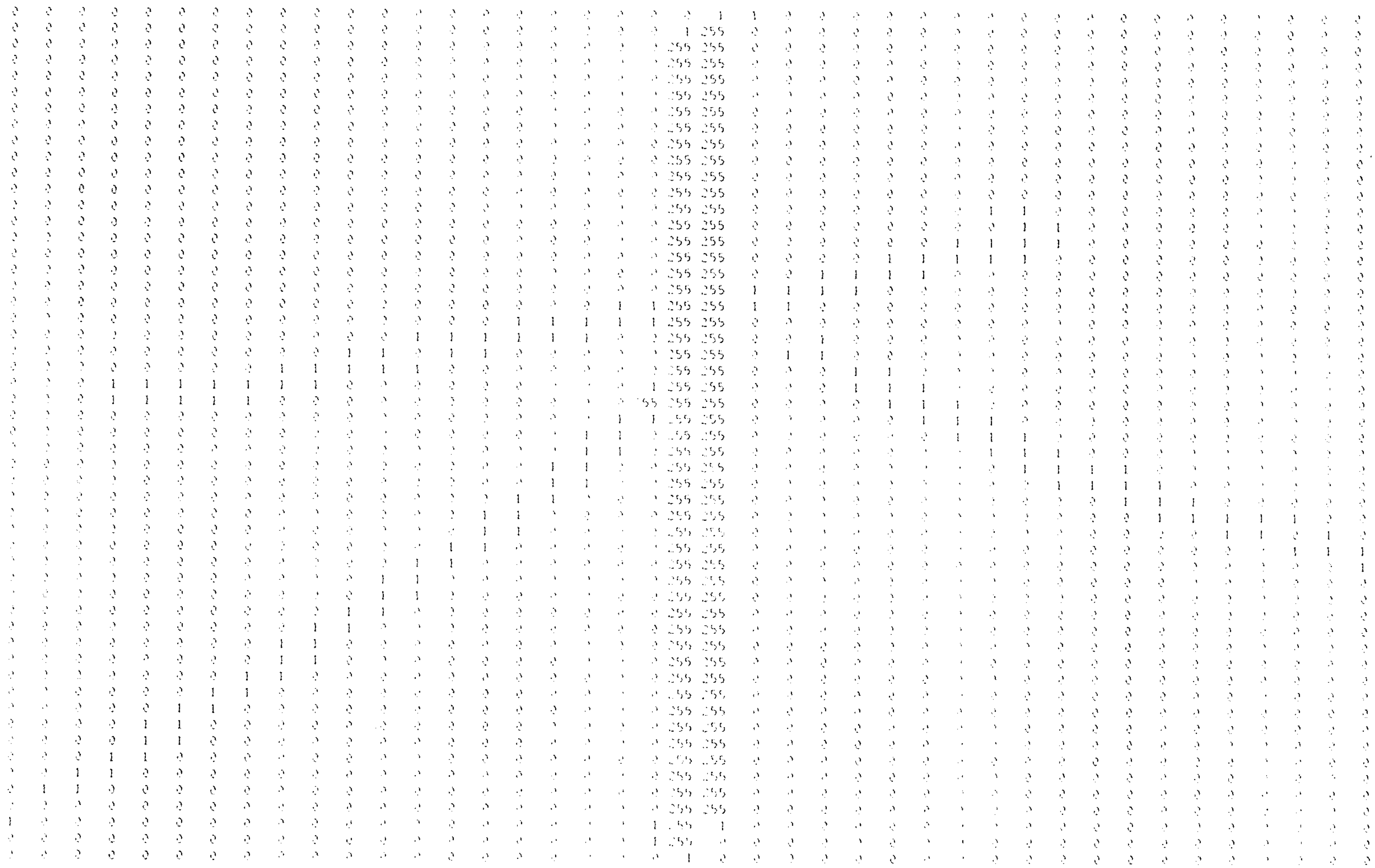

Figure 5 Satisfied result of vertical feature after post-processing

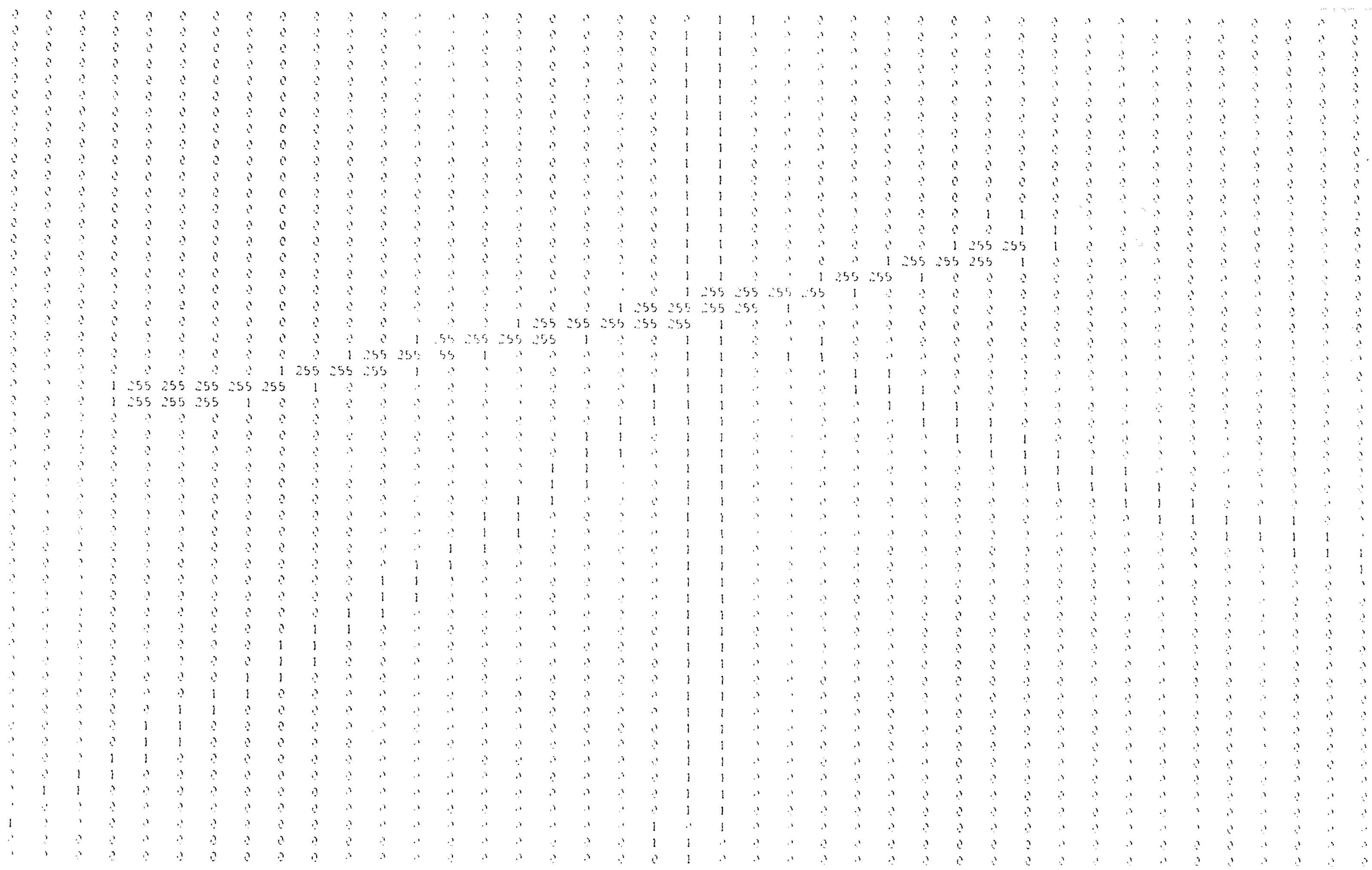

Figure 6 Satisfied result of horizontal feature after post-processing

3.6 Extraction of Slant Strokes

Extraction of slant strokes can be obtained from the original object excluded the vertical and horizontal strokes.

Algorithm of extraction of slant strokes

```
if ( (VerticalFeature[i][j]>1)  /* white object */
   ||(HorizontalFeature[i][j]>1)/* white object */
   ||(Result0[i][j]>1))       /* white background */
{
      OtherStroke[i][j]=0;  /* black background */
}
else
{
      OtherStroke[i][j]=255;   /* white object */
}
```

In addition to the post-processing of the deletion of single points, the breaking of the undesired connection, and the thresholding of stroke length, more procedures are required to seperate the slant strokes if they are connected together, and find out the direction of them. These procedures are also based on comparing of the coordinates of the top, bottom, left-most, and right-most end points of slant strokes.

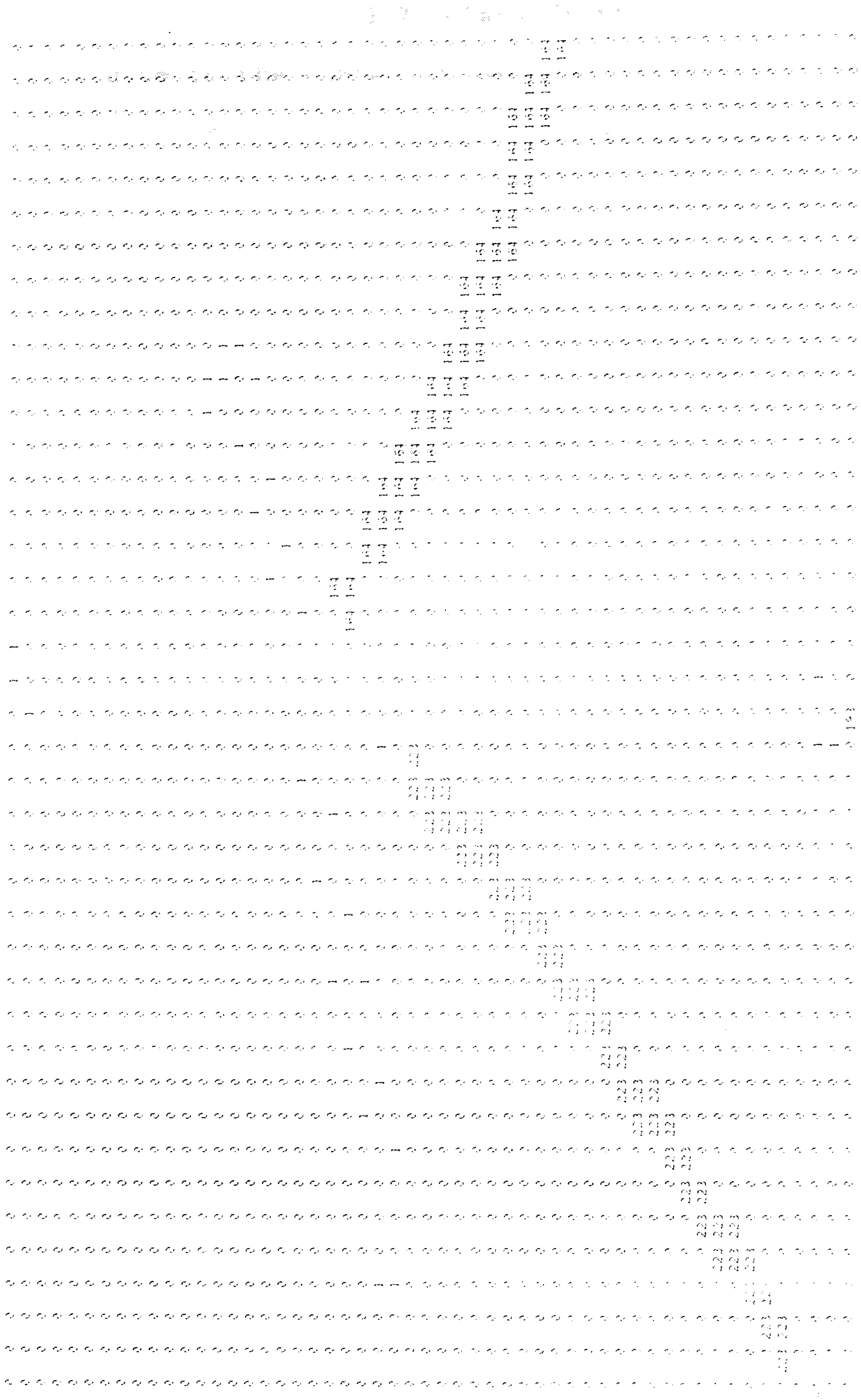

Figure 7 Two slant strokes extracted

3.7 Classification

According to the number of vertical, horizontal, and slant strokes, the Chinese character can be classified coarsely. A simple algorithm of classification is used here to verify that the feature extracted previously are enough for an efficient Chinese character recognition system. However, for the pratical application, this algorithm should be improved.

Algorithm of coarse classification:

```
switch (NumberOfVerticalStroke)
{
 ..
 case m:
 switch (NumberOfHorizotalStroke)
 {
  ..
  case n:
  switch (NumberOfSlantlStroke)
  {
   ..
   case r:
   Candidates of m vertical, n horizontal, and
                              r slant stroke
   Call fine-classification
   break;
   ..
   default:
```

```
    printf(" Not Defined ");
    break;
   }
   break;
   ..
   default:
   printf(" Not Defined ");
   break;
  }
  break;
  ..
  default:
  printf(" Not Defined ");
  break;
 }
```

It is possible that several similar characters would be classified to the same class. In this case, a fine classification based on the relative position of stroke would be required.

For example, the characters 七, 士, 土, 上, 工, 干, and 千 might be classified into the same class in the coarse classification due to having the same number of strokes, 1 vertical stroke, and 2 horizontal strokes. Fortunately, the relative position of these three strokes are different for each character. If not, they would even be impossibly distinguished by a Chinese.

Algorithms of fine classification are defined individually according to detail difference of the characters of the same coarse class.

Fine classification algorithm for 七, 士, 上, and 土:

```
if( LengthOfTopHorizontalLine <
     lengthOfBottomHorizontalLine )
  if ( LeftEndOfTopHorizontalLine < LeftEndOfVerticalLine)
  {
     return(" 土, Soil;");
  }
  else
  {
     return(" 上, Up;");
  }
else
  if ( LeftEndOfBottomHorizontalLine <
       LeftEndOfVerticalLIne )
  {
       return(" 士, Shii;");
  }
  else
  {
       return(" 七, 7;");
  }
```

CHAPTER 4

CONCLUSIONS

4.1 Results

Excluding loading the original document image onto the screen and the segmentation, the computing time for feature extraction, classifications, and one file I/O depends on the image size and stroke number of each Chinese character, and it is about 0.3 seconds (30 seconds for 100 characters) on an IBM PC 80486 DX33 and C language environment. The recognition rate is 96% without rejection. The error is caused by high distortion. The training and testing data are carefully selected 100 similar and not so easy to distinguish characters out of from 500 characters written by each of 15 persons of different ages from 9 to 63 years old, and education background from elementary school to graduate school. The character size may be from 18x18 to 60x60 pixels. The size of 40x40 pixels is suggested.

Compared to [7][8][9][10][11][12][13][14], the result of the proposed methods is outstanding.

4.2 Discussion

Comparing with [6][7], more resonable range is assigned to each kind of strokes, -40 to 40 degrees for the horizontal strokes, 40 to 70 degrees for slant strokes, and 70 to 110 degrees for the vertical strokes.

The proposed method records the top, bottom, leftmost, and rightmost ends of each stroke extracted, and then exact position and relation can be computed. That is why the very similar characters can be distinguished. It is also the reason that the thinning and normalization are not adopted because the detail information might be neglected.

The advantages of the proposed method include:

1. size invariance,
2. tolerance to certain distortion of strokes,
3. ideal time complexity,
4. strong ability to distinguish very similar characters, and
5. 96% high precision.(Note: This precision could be higher as long as more fine classification is introduced for some certain characters.)
6. Further, the rule developed for coarse and fine classification based on stroke matching in this thesis is suitable for on line hand-printed Chinese character rcognition, too.
7. The amazing performance of the proposed method is that it can endure the slope or rotation (in a certain degree) of the vertical and horizontal strokes.
8. The most important one is that it can detect the little difference among the very similar Chinese characters, e.g. 己, 已, and 巳.
9. Besides, the proposed method avoids the problems caused by thinning, normalization, and crossing.

However, the inflexibility for learning driven by the user is its disadvantage. This can also be overcome if an extra data base is built to store the learning result and act as the reserve for matching templete in case that the fine classification is not defined perfectly by the manufacturer.

4.3 Conclusions

Feature extraction based on simple traditional image processing skill is more preferred than that based on neural network methods on the pratical application of Optical Chinese Characters Recognition because it is expected that the parallel computers and languages are not as popular as personal computers, which is one kind of Von Neumann computers, in recent years.

Among so many Optical Chinese Characters Recognition based on traditional image processing skills, the proposed system has acceptable speed and ideal recognition rate.

4.4 Suggestion

The results for the sample data are acceptable, however, much more experiments for different characters are suggested since the total number of Chinese characters is so huge.

Besides, more efforts should be made on the storing, searching, and competing of the pattern base for matching to promote the performance of the whole recognition system.

APPENDIX

PROGRAM

```
/***********Vertion:117*********8,31***1994*************
    File     : TIF117.c   8:46
    Language: Turbo C
    Machine:  IBM PC AT 80486 DX 33
    Programmer: Sunshine Chang
    Master thesis of Mr. Sunshine Chang, 張善翔
 ********************************************************/
#include <io.h>
#include <mem.h>
#include <fcntl.h>
#include <string.h>
#include <graphics.h>
#include <dos.h>
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include "c:\tc\tif\ocr.lib"
void main (int argc, char *argv[])
{
   printf("%d", argc);
   strcpy(fname, argv[1]);
   open_file ();
   segmentation();
   feature();
```

```
   matching();
 }
/************ Vertion: 121 *****11,07***1994************
File     : ocr.lib   12:17     included by tif117.c
Researched & developed by Mr. Sunshine Chang, 張 善 翔
***************************** 山  不  在  高 **********/
unsigned int tti, bbi;
unsigned int llj, rrj;
unsigned int ti, tj;
unsigned int bi, bj;
unsigned int li, lj;
unsigned int ri, rj;
unsigned char grey5[64][64];
unsigned int iiii=63, jjjj=63;
unsigned int xk=257, yk=1150;
unsigned int vk=12, hk=19, sk=14;
unsigned int vertion=121;
unsigned char grey0[64][64];
unsigned char grey1[64][64];
unsigned char grey2[64][64];
unsigned char grey4[64][64];
void  PreProcess ()
{
  unsigned int i, j;
  float coef1[3*3] = {  0.0,   0.0,   0.0,
                        0.0,   1.0,   0.0,
                        0.0,  -1.0,   0.0  };
```

```
float coef2[3*3] = {  0.0,  0.0,  0.0,
                     -1.0,  1.0,  0.0,
                      0.0,  0.0,  0.0  };
float coef3[3*3] = {  0.0, -1.0,  0.0,
                      0.0,  1.0,  0.0,
                      0.0,  0.0,  0.0  };
float coef4[3*3] = {  0.0,  0.0,  0.0,
                      0.0,  1.0, -1.0,
                      0.0,  0.0,  0.0  };
     for (i = 0; i <= iiii; i++)
     {
        for (j = 0; j <= jjjj; j++)
        {
            if (*sp < threshold)
            {
              grey0[i][j] = 0;
            }
            else
            {
              grey0[i][j] = 255;
            }
            sp++;
        }
     }
     tti=iiii; bbi=0; llj=jjjj; rrj=0;
     for (i = 0; i <= iiii; i++)
     {
```

```
    for (j = 0; j <= jjjj; j++)
    {
         if(grey0[i][j]<1)
         {
           if(i<tti) tti=i;
           if(i>bbi) bbi=i;
           if(j<llj) llj=j;
           if(j>rrj) rrj=j;
         }
    }
}
for (i = tti+1; i <= bbi-1; i++)
{
    for (j = llj+1; j <= rrj-1; j++)
    {
         if(  (grey0[i-1][j]>1 )
            &&(grey0[i  ][j]<1 )
            &&(grey0[i+1][j]>1 ))
               grey0[i-1][j]=0;
         if(  (grey0[i][j-1]>1 )
            &&(grey0[i][j  ]<1 )
            &&(grey0[i][j+1]>1 ))
               grey0[i][j-1]=0;
    }
}
for (i = tti+1; i <= bbi-1; i++)
{
```

```
for (j = llj+1; j <= rrj-1; j++)
{
    grey1[i][j] =     coef1[0] * grey0[i-1][j-1]
                    + coef1[1] * grey0[i-1][j  ]
                    + coef1[2] * grey0[i-1][j+1]
                    + coef1[3] * grey0[i  ][j-1]
                    + coef1[4] * grey0[i  ][j  ]
                    + coef1[5] * grey0[i  ][j+1]
                    + coef1[6] * grey0[i+1][j-1]
                    + coef1[7] * grey0[i+1][j  ]
                    + coef1[8] * grey0[i+1][j+1];
    grey2[i][j] =     coef2[0] * grey0[i-1][j-1]
                    + coef2[1] * grey0[i-1][j  ]
                    + coef2[2] * grey0[i-1][j+1]
                    + coef2[3] * grey0[i  ][j-1]
                    + coef2[4] * grey0[i  ][j  ]
                    + coef2[5] * grey0[i  ][j+1]
                    + coef2[6] * grey0[i+1][j-1]
                    + coef2[7] * grey0[i+1][j  ]
                    + coef2[8] * grey0[i+1][j+1];
    grey3[i][j] =     coef3[0] * grey0[i-1][j-1]
                    + coef3[1] * grey0[i-1][j  ]
                    + coef3[2] * grey0[i-1][j+1]
                    + coef3[3] * grey0[i  ][j-1]
                    + coef3[4] * grey0[i  ][j  ]
                    + coef3[5] * grey0[i  ][j+1]
                    + coef3[6] * grey0[i+1][j-1]
```

```
                                 + coef3[7] * grey0[i+1][j  ]
                                 + coef3[8] * grey0[i+1][j+1];
                grey4[i][j] =      coef4[0] * grey0[i-1][j-1]
                                 + coef4[1] * grey0[i-1][j  ]
                                 + coef4[2] * grey0[i-1][j+1]
                                 + coef4[3] * grey0[i  ][j-1]
                                 + coef4[4] * grey0[i  ][j  ]
                                 + coef4[5] * grey0[i  ][j+1]
                                 + coef4[6] * grey0[i+1][j-1]
                                 + coef4[7] * grey0[i+1][j  ]
                                 + coef4[8] * grey0[i+1][j+1];
            }
        }
}
PreProcess2()
{
  unsigned int i, j;
        for (i = tti+1; i <= bbi-1; i++)
        {
            for (j = llj+1; j <= rrj-1; j++)
            {
                if(  (grey5[i-1][j-1] <=1 )
                   &&(grey5[i  ][j-1] <=1 )
                   &&(grey5[i+1][j-1] <=1 )
                   &&(grey5[i-1][j  ] <=1 )
                   &&(grey5[i  ][j  ]  >1 )
                   &&(grey5[i+1][j  ] <=1 )
```

```
              &&(grey5[i-1][j+1] <=1 )
              &&(grey5[i  ][j+1] <=1 )
              &&(grey5[i+1][j+1] <=1 ))
                 grey5[i][j] = 0;
          if(  (grey6[i-1][j-1] <=1 )
              &&(grey6[i  ][j-1] <=1 )
              &&(grey6[i+1][j-1] <=1 )
              &&(grey6[i-1][j  ] <=1 )
              &&(grey6[i  ][j  ]  >1 )
              &&(grey6[i+1][j  ] <=1 )
              &&(grey6[i-1][j+1] <=1 )
              &&(grey6[i  ][j+1] <=1 )
              &&(grey6[i+1][j+1] <=1 ))
                 grey6[i][j] = 0;
     }
}
for (i=tti; i<=bbi; i++)
{
     for (j=llj+3; j<=rrj; j++)
     {
          if(  (grey5[i][j-3] >1 )
              &&(grey5[i][j-2] >1 )
              &&(grey5[i][j-1] >1 )
              &&(grey5[i][j  ] >1 ))
                 grey5[i][j  ] =1;
     }
}
```

```
for (i=tti+3; i<=bbi; i++)
{
    for (j=llj; j<=rrj; j++)
    {
        if(  (grey6[i-3][j] >1 )
           &&(grey6[i-2][j] >1 )
           &&(grey6[i-1][j] >1 )
           &&(grey6[i  ][j] >1 ))
              grey6[i  ][j] =1;
    }
}
for (i=tti+3; i<=bbi; i++)
{
    for (j=llj+3; j<=rrj; j++)
    {
        if(  (grey5[i-3][j-3] >1 )
           &&(grey5[i-2][j-2] >1 )
           &&(grey5[i-1][j-1] >1 )
           &&(grey5[i  ][j  ] >1 ))
              grey5[i  ][j  ] =1;
        if(  (grey6[i-3][j-3] >1 )
           &&(grey6[i-2][j-2] >1 )
           &&(grey6[i-1][j-1] >1 )
           &&(grey6[i  ][j  ] >1 ))
              grey6[i  ][j  ] =1;
    }
}
```

```
for (i = tti; i <= bbi-3; i++)
{
    for (j = llj+3; j <= rrj; j++)
    {
        if(  (grey5[i+3][j-3] >1 )
           &&(grey5[i+2][j-2] >1 )
           &&(grey5[i+1][j-1] >1 )
           &&(grey5[i  ][j  ] >1 ))
              grey5[i  ][j  ] =1;
        if(  (grey6[i+3][j-3] >1 )
           &&(grey6[i+2][j-2] >1 )
           &&(grey6[i+1][j-1] >1 )
           &&(grey6[i  ][j  ] >1 ))
              grey6[i  ][j  ] =1;
    }
}
for (i = tti; i <= bbi-3; i++)
{
    for (j = llj+1; j <= rrj-6; j++)
    {
        if(  (grey6[i  ][j-1] >1 )
           &&(grey6[i  ][j  ] >1 )
           &&(grey6[i+1][j+1] >1 )
           &&(grey6[i+1][j+2] >1 )
           &&(grey6[i+2][j+3] >1 )
           &&(grey6[i+2][j+4] >1 )
           &&(grey6[i+3][j+5] >1 )
```

```
                &&(grey6[i+3][j+6] >1 ))
                {
                   grey6[i  ][j  ] =1;
                   grey6[i  ][j-1] =1;
                }
            }
        }
}
void  Extraction(unsigned int i, unsigned int j)
{
            if ((grey1[i][j]==1)||(grey3[i][j]==1))
            {
               grey5[i][j]=1;
            }
            else if(grey0[i][j]>0)
            {
               grey5[i][j]=0;
            }
            else
            {
               grey5[i][j]=255;
            }
            if ((grey2[i][j]==1)||(grey4[i][j]==1))
            {
               grey6[i][j]=1;
            }
            else if(grey0[i][j]>0)
```

```
            {
               grey6[i][j]=0;
            }
            else
            {
               grey6[i][j]=255;
            }
}
void  dN4SingleP()
{
       unsigned int i, j;
       for (i=tti+1; i<=bbi-1; i++)
       {
           for (j=llj+1; j<=rrj-1; j++)
           {
               if(/*   (grey3[i-1][j-1] <=1 )    */
                       (grey3[i  ][j-1] <=1 )
                  /* &&(grey3[i+1][j-1] <=1 )    */
                     &&(grey3[i-1][j  ] <=1 )
                     &&(grey3[i  ][j  ]  >1 )
                     &&(grey3[i+1][j  ] <=1 )
                  /* &&(grey3[i-1][j+1] <=1 )    */
                     &&(grey3[i  ][j+1] <=1 )
                  /* &&(grey3[i+1][j+1] <=1 )    */ )
                        grey3[i][j] = 1;
           }
       }
```

```
void feature ()
    {
      PreProcess();
      j=0;
      fprintf(out,"    ");
      for (i=llj; i<=rrj; i++)
      {
        fprintf(out,"%d", j);
        j++;
        if(j>=10) j=0;
      }
      fprintf(out,"\n");
      k=0;
      for (i=tti; i<=bbi; i++)
      {
         fprintf(out,"\n%d  ", k);
         for (j = llj; j <= rrj; j++)
         {
            if(grey0[i][j]==255)
               fprintf(out,".");
            else if(grey0[i][j]==1)
               fprintf(out,"_");
            else
               fprintf(out,"#");
            Extraction(i, j);
         }
         fprintf(out,"  %d", k);
```

```
      k++;
      if(k>=10) k=0;
   }
   fprintf(out,"\n\n   ");
   j=0;
   for (i=llj; i<=rrj; i++)
   {
     fprintf(out,"%d", j);
     j++;
     if(j>=10) j=0;
   }
   fprintf(out,"\n\n\n%3d %3d %3d %3d\n%s %3d %3d %3d
           %3d %3d %3d\n%d %s",tti-tti,llj-llj,bbi-
           tti,rrj-llj,fname,xk,yk,gk,(bbi-tti+rrj-
           llj)/vk,(bbi-tti+rrj-llj)/hk,(bbi-tti+rrj-
           llj)/sk, vertion, bname);
   fclose(out);
   PreProcess2();
   for (i = tti; i <= bbi; i++)
     for (j = llj; j <= rrj; j++)
       grey1[i][j] =0;
   strcpy(bname, aname);
   strcat(bname, ".8");
   out8 = fopen(bname, "w");
   fprintf(out8,"\n\n   Class  |      Top |   Bottom |
           Left-most|Right-most|");
   code = 254;
```

```
count = 0;
for (i=tti; i<=bbi; i++)
{
     for (j=llj; j<=rrj; j++)
     {
          if ( grey5[i][j] >= 255 )
          {
            ii = i;  jj = j;
            ti = i;  tj = j;
            bi = i;  bj = j;
            lj = j;  li = i;
            rj = j;  ri = i;
            mark5(code, ii, jj);
            if(((bi-ti)<=3)||
              ((bi-ti)<=((bbi-tti)+(rrj-llj))/vk))
            {
              erase5(code, ii, jj);
            }
            else
            {
              if (code < 113)
              {
                count = count + 1;
                code = 254 - count;
              }
              code = code - 30;
            }
```

```
            }
        }
    }
    code = 254;
    count = 0;
    for (i=tti; i<=bbi; i++)
    {
        for (j=llj; j<=rrj; j++)
        {
            if ( grey6[i][j] >= 255 )
            {
              ii = i;  jj = j;
              ti = i;  tj = j;
              bi = i;  bj = j;
              lj = j;  li = i;
              rj = j;  ri = i;
              mark6(code, ii, jj);
              if (((rj-lj)<=3)||
              ((rj-lj)<=((bbi-tti)+(rrj-llj))/hk))
                erase6(code, ii, jj);
              else
              {
                if (code < 113)
                {
                  count = count + 1;
                  code = 254 - count;
                }
```

```
                code = code - 30;
              }
            }
        }
    }
    strcpy(bname, aname);
    strcat(bname, ".5");
    out5 = fopen(bname, "w");
    strcpy(bname, aname);
    strcat(bname, ".6");
    out6 = fopen(bname, "w");
    j=0;
    for (i=llj; i<=rrj; i++)
    {
      fprintf(out5,"%d", j);
      fprintf(out6,"%d", j);
      j++;
      if(j>=10) j=0;
    }
    fprintf(out5,"\n");
    fprintf(out6,"\n");
    k=0;
    for (i=tti; i<=bbi; i++)
    {
       fprintf(out5,"\n%d  ", k);
       fprintf(out6,"\n%d  ", k);
       for (j=llj; j<=rrj; j++)
```

```
{
   if (grey5[i][j]==1)
   {
      fprintf(out5,"_");
   }
   else if(grey5[i][j]==0)
   {
      fprintf(out5,".");
   }
   else
   {
      fprintf(out5,"V");
   }
   if (grey6[i][j]==1)
   {
      fprintf(out6,"_");
   }
   else if(grey6[i][j]==0)
   {
      fprintf(out6,".");
   }
   else
   {
      fprintf(out6,"H");
   }
   if ((grey5[i][j]>1)||
        (grey6[i][j]>1)||(grey0[i][j]>1))
```

```
        {
           grey3[i][j]=0;
        }
        else
        {
           grey3[i][j]=255;
        }
     }
     fprintf(out5,"  %d", k);
     fprintf(out6,"  %d", k);
     k++;
     if(k>=10) k=0;
  }
  fprintf(out5,"\n\n   ");
  fprintf(out6,"\n\n   ");
  j=0;
  for (i=llj; i<=rrj; i++)
  {
    fprintf(out5,"%d", j);
    fprintf(out6,"%d", j);
    j++;
    if(j>=10) j=0;
  }
  fprintf(out5,"\n\n\n%3d %3d %3d %3d\n%s %3d %3d %3d
          %3d %3d %3d\n%d %s", tti-tti,llj-llj,bbi-
          tti,rrj-llj,fname,xk,yk,gk,(bbi-tti+rrj-
          llj)/vk,(bbi-tti+rrj-llj)/hk,(bbi-tti+rrj-
```

```
        llj)/sk, vertion, bname);
fclose(out5);
fprintf(out6,"\n\n\n%3d %3d %3d %3d\n%s %3d %3d %3d
        %3d %3d %3d\n%d %s", tti-tti,llj-llj,bbi-
        tti,rrj-llj,fname,xk,yk,gk,(bbi-tti+rrj-
        llj)/vk,(bbi-tti+rrj-llj)/hk,(bbi-tti+rrj-
        llj)/sk, vertion, bname);
fclose(out6);
dN4SingleP();
code = 254;
count = 0;
for (i=tti; i<=bbi; i++)
{
    for (j=llj; j<=rrj; j++)
    {
        if ( grey3[i][j] >= 255 )
        {
          ii = i;  jj = j;
          ti = i;  tj = j;
          bi = i;  bj = j;
          lj = j;  li = i;
          rj = j;  ri = i;
          mark7(code, ii, jj);
          if (((bi-ti)<5)||
             ((bi-ti)<=((bbi-tti)+(rrj-llj))/sk))
          {
            erase7(code, ii, jj);
```

```
                }
                else
                {
                  if (code < 113)
                  {
                    count = count + 1;
                    code = 254 - count;
                  }
                  code = code - 30;
                }
              }
          }
      }
   match();
     fclose(out8);
      strcpy(bname, aname);
      strcat(bname, ".7");
      out7 = fopen(bname, "w");
      j=0;
      fprintf(out7,"    ");
      for (i=llj; i<=rrj; i++)
      {
        fprintf(out7,"%d", j);
        j++;
        if(j>=10) j=0;
      }
      fprintf(out7,"\n");
```

```
k=0;
for (i=tti; i<=bbi; i++)
{
    fprintf(out7,"\n%d  ", k);
    for (j=llj; j<=rrj; j++)
    {
        if (grey3[i][j]==1)
        {
            fprintf(out7,"_");
        }
        else if(grey3[i][j]==0)
        {
            fprintf(out7,".");
        }
        else
        {
            fprintf(out7,"X");
        }
    }
    fprintf(out7,"  %d", k);
    k++;
    if(k>=10) k=0;
}
fprintf(out7,"\n\n   ");
j=0;
for (i=llj; i<=rrj; i++)
{
```

```
        fprintf(out7,"%d", j);
        j++;
        if(j>=10) j=0;
      }
      fclose(out7);
}
```

APPENDIX B

SOME RESULTS

Example 1:

```
   0123456789012345678901234567890123456789

0  ................##......................  0
1  ................##......................  1
2  .................##.....................  2
3  .................##.....................  3
4  .................##.....................  4
5  .................##...................#.  5
6  .................##...............#####.  6
7  .................##.............######..  7
8  .................##..........######.....  8
9  ..................##.......#####........  9
0  ..................##....#####...........  0
1  ..................##..#####.............  1
2  ..................######................  2
3  ................######..................  3
4  ..............######....................  4
5  ............####..##....................  5
6  .........#####....##....................  6
7  .......#####......##....................  7
8  ....#####.........##....................  8
9  .######............##...................  9
0  ####...............##...................  0
1  #..................##...................  1
2  ...................##...............####  2
3  ....................##.......###########  3
4  ....................##...###########....  4
5  .....................########...........  5
6  ......................###...............  6

   0123456789012345678901234567890123456789
```

CoarseClassificationResult===> 七, 士, 工, 上, 土

FineClassificationResult---> 七

Vertical feature, represented by symbol V, of example 1

```
   0123456789012345678901234567890123456789

0  ................VV......................  0
1  ................_V......................  1
2  .................V_.....................  2
3  .................VV.....................  3
4  .................VV.....................  4
5  .................VV..................._.  5
6  .................VV..............._____.  6
7  .................VV.............__.___..  7
8  ................._V.........______.....  8
9  ..................V_......._____........  9
0  ..................VV...._____...........  0
1  ..................VV.._____.............  1
2  ..................VV____................  2
3  ................__VV__..................  3
4  ..............____VV....................  4
5  ............____..VV....................  5
6  ........._____....VV....................  6
7  ......._____......VV....................  7
8  ....._____........._V....................  8
9  ._______.............V_...................  9
0  ____...............VV...................  0
1  _..................VV...................  1
2  ..................._V..............____  2
3  ....................V_......___________  3
4  ...................._V..___________....  4
5  .....................________...........  5
6  ......................___...............  6

   0123456789012345678901234567890123456789
```

Horizontal feature, represented by symbol H, of example 1

```
     0123456789012345678901234567890123456789

0    ................__......................   0
1    ................__......................   1
2    .................__.....................   2
3    .................__.....................   3
4    .................__.....................   4
5    .................__..................._.   5
6    .................__..............._HHH_.   6
7    .................__............._HHHH_..   7
8    .................__.........._HHHH_.....   8
9    ..................__......._HHH_........   9
0    ..................__...._HHH_...........   0
1    ..................__.._HHH_.............   1
2    .................._HHHH_................   2
3    ................_HHHH_..................   3
4    .............._HHHH_....................   4
5    ............_HH_..__....................   5
6    ........._HHH_....__....................   6
7    ......._HHH_......__....................   7
8    ...._HHH_.........__....................   8
9    ._HHHH_............__...................   9
0    HHH_...............__...................   0
1    H..................__...................   1
2    ...................__..............._HHH   2
3    ....................__......._HHHHHHHHHH   3
4    ....................__..._HHHHHHHHH_....   4
5    ....................._HHHHHH_...........   5
6    ......................HHH...............   6

     0123456789012345678901234567890123456789
```

No slant feature of example 1

```
   0123456789012345678901234567890123456789

0  ........................................  0
1  ................_.......................  1
2  .................._.....................  2
3  ........................................  3
4  ........................................  4
5  ......................................_.  5
6  .................................._..._.  6
7  ................................_...._..  7
8  ................._..........._...._.....  8
9  ..................._......._..._........  9
0  ........................_..._...........  0
1  ......................_..._.............  1
2  ......................._................  2
3  ................_...._..................  3
4  .............._.........................  4
5  ............_.._........................  5
6  ........._..._..........................  6
7  ......._..._............................  7
8  ...._..._........._.....................  8
9  ._...._............._...................  9
0  ..._....................................  0
1  ........................................  1
2  ..................._................_...  2
3  ....................._......._..........  3
4  ...................._...._........._....  4
5  ....................._......_...........  5
6  ........................................  6

   0123456789012345678901234567890123456789
```

Example 2:

```
   0123456789012345678901234567890123456789

0  ...................#....................  0
1  ..................##....................  1
2  ..................###...................  2
3  ...................##...............##.#  3
4  ...................##.........#.#...##..  4
5  ...................##......######.......  5
6  ...................###########..........  6
7  .................##########.............  7
8  ...........##########...................  8
9  ...#...##########.###...................  9
0  #..########........##...................  0
1  #...###...........##....................  1
2  ..................##....................  2
3  ..................##....................  3
4  ..................##....................  4
5  ..................##....................  5
6  .................##.....................  6
7  .................##.....................  7
8  .................##.....................  8
9  .................##.....................  9
0  ........................................  0
1  ................##......................  1
2  ................##......................  2
3  ........................................  3
4  .......................#####............  4
5  ..................##########............  5
6  ...............########.................  6
7  ...........#######......................  7
8  ..........#####.........................  8
9  ..........#.............................  9

   0123456789012345678901234567890123456789
```

CoarseClassificationResult===> 七, 士, 工, 上, 土

FineClassificationResult---> 士

Vertical feature, represented by symbol V, of example 2

```
     0123456789012345678901234567890123456789

0    ...................V....................    0
1    .................._V....................    1
2    .................._V_...................    2
3    ...................VV...............__._    3
4    ...................VV........._._...__..    4
5    ...................VV......______.......    5
6    ...................VV_________..........    6
7    .................__VV______.............    7
8    ..........._______VVV...................    8
9    ..._...__________._VV...................    9
0    _..________........V_...................    0
1    _...___..........._V....................    1
2    ..................VV....................    2
3    ..................VV....................    3
4    ..................VV....................    4
5    ..................V_....................    5
6    ................._V.....................    6
7    .................VV.....................    7
8    .................VV.....................    8
9    .................__.....................    9
0    ........................................    0
1    ................__......................    1
2    ................__......................    2
3    ........................................    3
4    ......................._____............    4
5    ..................__________............    5
6    ...............________.................    6
7    ..........._______......................    7
8    ..........._____........................    8
9    .........._.............................    9

     0123456789012345678901234567890123456789
```

Horizontal feature, represented by symbol H, of example 2

```
     0123456789012345678901234567890123456789

0    ..................._....................   0
1    ..................__....................   1
2    .................._._...................   2
3    ...................__...............__._   3
4    ...................__........._._...__..   4
5    ...................__......_HHHH_.......   5
6    ..................._HHHHHHHHH_..........   6
7    ................._HHHHHHHH_.............   7
8    ..........._HHHHHHHH_...................   8
9    ..._..._HHHHHHHH_._H_...................   9
0    _.._HHHHHH_........__...................   0
1    _..._H_...........__....................   1
2    ..................__....................   2
3    ..................__....................   3
4    ..................__....................   4
5    ..................__....................   5
6    .................__.....................   6
7    .................__.....................   7
8    .................__.....................   8
9    .................__.....................   9
0    ........................................   0
1    ................__......................   1
2    ................__......................   2
3    ........................................   3
4    ......................._HHH_............   4
5    .................._HHHHHHHH_............   5
6    ..............._HHHHHH_.................   6
7    ..........._HHHHH_......................   7
8    .........._HHH_.........................   8
9    ..........H.............................   9

     0123456789012345678901234567890123456789
```

No slant feature of example 2

```
    0123456789012345678901234567890123456789

0   ........................................   0
1   .................._.....................   1
2   .................._._...................   2
3   ....................................__.X   3
4   .............................._._...__..   4
5   ..........................._...._.......   5
6   ............................._..........   6
7   ................._........_.............   7
8   ..........._............................   8
9   ..._..._........_._.....................   9
0   _.._......_........._...................   0
1   _..._._..........._.....................   1
2   ........................................   2
3   ........................................   3
4   ........................................   4
5   ..................._....................   5
6   ................._......................   6
7   ........................................   7
8   ........................................   8
9   .................__.....................   9
0   ........................................   0
1   ................__......................   1
2   ................__......................   2
3   ........................................   3
4   ......................._..._............   4
5   .................._........_............   5
6   ..............._......_.................   6
7   ..........._....._......................   7
8   .........._..._.........................   8
9   ........................................   9

    0123456789012345678901234567890123456789
```

Example 3:

```
    01234567890123456789012345678901234567

0   ...................##...........................   0
1   ...................####.........................   1
2   ...................####.........................   2
3   ...................##...........................   3
4   ...................##...........................   4
5   ...................###..........................   5
6   ...................###............#.............   6
7   ....................################............   7
8   ..................##################............   8
9   ...............###########......................   9
0   .....#.############.###.........................   0
1   ...#############....###.........................   1
2   ...##.####..........###.........................   2
3   ....................##..........................   3
4   ....................##..........................   4
5   ....................##..........................   5
6   ....................##..........................   6
7   ....................##..........................   7
8   ....................##..........................   8
9   ...................##...........................   9
0   ...................##...........................   0
1   ...................##...........................   1
2   ...................##...........................   2
3   ..................##............................   3
4   ..................##............................   4
5   ..................##............................   5
6   .................##.............................   6
7   .................##.............................   7
8   .................##.............................   8
9   ................................................   9
0   ...........................................###..   0
1   .....................................##########.   1
2   ..............................#############...##   2
3   ......................###############...........   3
4   ...................###########..................   4
5   ...............########.........................   5
6   .###################............................   6
7   ################................................   7
8   .##########.....................................   8

    01234567890123456789012345678901234567
```

CoarseClassificationResult===> 七, 士, 工, 上, 土

FineClassificationResult---> 土

Vertical feature, represented by symbol V, of example 3

```
   012345678901234567890123456789012345678901234567

0  ...................VV...........................  0
1  ...................VV__.........................  1
2  ...................VV__.........................  2
3  ...................VV...........................  3
4  ...................VV...........................  4
5  ...................VV_..........................  5
6  ..................._VV............_.............  6
7  ....................VV____________._............  7
8  ..................__VV______________............  8
9  ...............___._VVV___......................  9
0  ....._.________.___.VVV.........................  0
1  ..._____________....VVV.........................  1
2  ...__.____..........VV_.........................  2
3  ....................VV..........................  3
4  ....................VV..........................  4
5  ....................VV..........................  5
6  ....................VV..........................  6
7  ....................VV..........................  7
8  ....................V_..........................  8
9  ..................._V...........................  9
0  ...................VV...........................  0
1  ...................VV...........................  1
2  ...................V_...........................  2
3  .................._V............................  3
4  ..................VV............................  4
5  ..................V_............................  5
6  ................._V.............................  6
7  .................VV.............................  7
8  .................__.............................  8
9  ................................................  9
0  ...........................................___..  0
1  ....................................._________.  1
2  .............................._____________...__  2
3  ......................_______________...........  3
4  ...................___._______..................  4
5  ...............____.___.........................  5
6  .______________.____............................  6
7  ________________................................  7
8  .__________.....................................  8

   012345678901234567890123456789012345678901234567
```

Horizontal feature, represented by symbol H, of example 3

```
   012345678901234567890123456789012345678901234567

0  ...................__...........................  0
1  ...................____.........................  1
2  ...................____.........................  2
3  ...................__...........................  3
4  ...................__...........................  4
5  ..................._H_..........................  5
6  ..................._H_............_.............  6
7  ...................._HHHHHHHHHHHHHH_............  7
8  .................._HHHHHHHHHHHHHHHH_............  8
9  ..............._HHHHHHH_H_......................  9
0  ....._._HHHHHHHHHH_.___.........................  0
1  ..._HHHHHHHHHHH_....___.........................  1
2  ...__._HH_..........___.........................  2
3  ....................__..........................  3
4  ....................__..........................  4
5  ....................__..........................  5
6  ....................__..........................  6
7  ....................__..........................  7
8  ....................__..........................  8
9  ...................__...........................  9
0  ...................__...........................  0
1  ...................__...........................  1
2  ...................__...........................  2
3  ..................__............................  3
4  ..................__............................  4
5  ..................__............................  5
6  .................__.............................  6
7  .................__.............................  7
8  .................__.............................  8
9  ................................................  9
0  ..........................................._H_..  0
1  ....................................._HHHHHHHH_.  1
2  .............................._HHHHHHHHHHH_...__  2
3  ......................_HHHHHHHHHHHHH_...........  3
4  ..................._HHHHHHHHH_..................  4
5  ..............._HHHHHH_.........................  5
6  ._HHHHHHHHHHHHHHHHH_............................  6
7  HHHHHHHHHHHHHHH_................................  7
8  .HHHHHHHHHH.....................................  8

   012345678901234567890123456789012345678901234567
```

No slant feature of example 3

```
   012345678901234567890123456789012345678901234567

0  ................................................  0
1  .....................__.........................  1
2  .....................__.........................  2
3  ................................................  3
4  ................................................  4
5  ....................._..........................  5
6  ..................._.............._.............  6
7  ..................................._............  7
8  .................._................_............  8
9  ..............._......._._......................  9
0  ....._._.........._.............................  0
1  ..._..........._................................  1
2  ...__._.._............_.........................  2
3  ................................................  3
4  ................................................  4
5  ................................................  5
6  ................................................  6
7  ................................................  7
8  ....................._..........................  8
9  ..................._............................  9
0  ................................................  0
1  ................................................  1
2  ...................._...........................  2
3  .................._.............................  3
4  ................................................  4
5  ..................._............................  5
6  ................._..............................  6
7  ................................................  7
8  .................__.............................  8
9  ................................................  9
0  ..........................................._._..  0
1  ....................................._........_.  1
2  .............................._..........._..._X  2
3  ......................_............._........._.  3
4  ..................._........._..................  4
5  ..............._......_.........................  5
6  ._................._............................  6
7  ..............._................................  7
8  ................................................  8

   012345678901234567890123456789012345678901234567
```

Example 4:

```
   012345678901234567890123456789012345678901234567890

0  ..................##...............................  0
1  ..................###..............................  1
2  ..................###..............................  2
3  ..................##...............................  3
4  ...................##..............................  4
5  ...................##..............................  5
6  ...................##..............................  6
7  ...................##..............................  7
8  ...................##..................##..........  8
9  ...................##...............#####..........  9
0  ...................##............######............  0
1  ...................##.........#######..............  1
2  ...................###...#########.................  2
3  ...................###.#.#####.....................  3
4  ....................####...........................  4
5  ....................##.............................  5
6  ....................##.............................  6
7  ....................##.............................  7
8  ....................##.............................  8
9  ....................##.............................  9
0  ....................##.............................  0
1  ....................##.............................  1
2  ....................##.............................  2
3  ....................##.............................  3
4  ....................##.............................  4
5  ....................##.............................  5
6  ....................##.............................  6
7  ....................##.............................  7
8  ....................##.............................  8
9  ....................##.............................  9
0  ....................##.............................  0
1  ....................##.....................#######.  1
2  ....................##............#################  2
3  ....................##..###################........  3
4  ...................###############.................  4
5  ...........#############...........................  5
6  ....################...............................  6
7  ###########........................................  7
8  .####..............................................  8

   012345678901234567890123456789012345678901234567890
```

CoarseClassificationResult===> 七, 士, 工, 上, 土

FineClassificationResult---> 上

Vertical feature, represented by symbol V, of example 4

```
     012345678901234567890123456789012345678901234567890

0    ..................VV...............................    0
1    ..................VV_..............................    1
2    ..................VV_..............................    2
3    .................._V...............................    3
4    ...................V_..............................    4
5    ...................VV..............................    5
6    ...................VV..............................    6
7    ...................VV..............................    7
8    ...................VV..................__..........    8
9    ...................VV..............._____..........    9
0    ...................VV............___.__............    0
1    ...................VV........._____.___..............  1
2    ...................VV_..._________.................    2
3    ..................._VV._._____.....................    3
4    ....................VV__...........................    4
5    ....................VV.............................    5
6    ....................VV.............................    6
7    ....................VV.............................    7
8    ....................VV.............................    8
9    ....................VV.............................    9
0    ....................VV.............................    0
1    ....................VV.............................    1
2    ....................VV.............................    2
3    ....................VV.............................    3
4    ....................VV.............................    4
5    ....................VV.............................    5
6    ....................VV.............................    6
7    ....................VV.............................    7
8    ....................VV.............................    8
9    ....................VV.............................    9
0    ....................VV.............................    0
1    ....................VV....................._______.    1
2    ....................VV............_________________    2
3    ....................VV..___________________........    3
4    ..................._VV____________.................    4
5    ...........________V____...........................    5
6    ....________________...............................    6
7    ___________........................................    7
8    .____..............................................    8

     012345678901234567890123456789012345678901234567890
```

Horizontal feature, represented by symbol H, of example 4

```
    012345678901234567890123456789012345678901234567890

0   ..................__...............................   0
1   ..................___..............................   1
2   ..................___..............................   2
3   ..................__...............................   3
4   ...................__..............................   4
5   ...................__..............................   5
6   ...................__..............................   6
7   ...................__..............................   7
8   ...................__..................__..........   8
9   ...................__..............._HHH_..........   9
0   ...................__............_HHHH_............   0
1   ...................__........._HHHHH_..............   1
2   ...................___..._HHHHHHH_.................   2
3   ...................___._._HHH_.....................   3
4   ....................____...........................   4
5   ....................__.............................   5
6   ....................__.............................   6
7   ....................__.............................   7
8   ....................__.............................   8
9   ....................__.............................   9
0   ....................__.............................   0
1   ....................__.............................   1
2   ....................__.............................   2
3   ....................__.............................   3
4   ....................__.............................   4
5   ....................__.............................   5
6   ....................__.............................   6
7   ....................__.............................   7
8   ....................__.............................   8
9   ....................__.............................   9
0   ....................__.............................   0
1   ....................__....................._HHHHH_.   1
2   ....................__............_HHHHHHHHHHHHHHHH   2
3   ....................__.._HHHHHHHHHHHHHHHHH_........   3
4   ..................._HHHHHHHHHHHHH_.................   4
5   ..........._HHHHHHHHHHH_...........................   5
6   ...._HHHHHHHHHHHHHH_...............................   6
7   HHHHHHHHHH_........................................   7
8   .HHHH..............................................   8

    012345678901234567890123456789012345678901234567890
```

No slant feature of example 4

```
  012345678901234567890123456789012345678901234567890

0 ................................................... 0
1 ...................._.............................. 1
2 ...................._.............................. 2
3 .................._................................ 3
4 ...................._.............................. 4
5 ................................................... 5
6 ................................................... 6
7 ................................................... 7
8 .......................................__.......... 8
9 ...................................._..._.......... 9
0 ................................._...._............ 0
1 .............................._....._.............. 1
2 ....................._..._......._................. 2
3 ..................._..._._..._..................... 3
4 ......................__........................... 4
5 ................................................... 5
6 ................................................... 6
7 ................................................... 7
8 ................................................... 8
9 ................................................... 9
0 ................................................... 0
1 ................................................... 1
2 ................................................... 2
3 ................................................... 3
4 ................................................... 4
5 ................................................... 5
6 ................................................... 6
7 ................................................... 7
8 ................................................... 8
9 ................................................... 9
0 ................................................... 0
1 ..........................................._....._. 1
2 .................................._................ 2
3 ........................_................._........ 3
4 ..................._............._................. 4
5 ..........._..........._........................... 5
6 ...._.............._............................... 6
7 .........._........................................ 7
8 ................................................... 8

  012345678901234567890123456789012345678901234567890
```

REFERENCES

[1] Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing*, Addison Wesley Pub. Co., New York, 1992.

[2] Steve Rimmer, *Bit Mapped Image*, Blue Ridge Summit, PA: Winderest, 1990.

[3] K.Y. Wong, R.G.Casey, and F.H.Whal, "Document Analysis System," IBM J. Res. Develop., Vol 26, No.6, 1982, 647-656.

[4] F.M. Wahl, K.Y. Wong, and R.G. Casey "Block Segmentation and Text Extraction in Mixed Text/Image Documents," CGIP, Vol. 20,1982, 357-390.

[5] Roger L.T. Cederberg, "Chain-Link Coding and Segment for Raster Scan Devices," CGIP, Vol.10 1979,p224-234.

[6] Geeng-How Chang, "An Accumulated and Learnable Multifont OCCR," Masters Thesis, National Taiwan University, Taiwan, R.O.C., 1990.

[7] Ding-Long Chang, "Strokes Extraction of Chinese Characters," Masters Thesis, Tamkang University, Taiwan, R.O.C., 1991.

[8] Hung-You Chao, "A New Stroke Extraction Method for Hand-Written Characters," Masters Theis, National Chiao Tung University, Taiwan, R.O.C., 1993.

[9] Lee-Hung Cho, "Language Model in the OCCR," Masters Thesis, National Hsing Hua University, Taiwan, R.O.C., 1992.

[10] Che-Hui Chang Chien, "A Markov Language Model in Handwritten Chinese Text Recognition," Masters Thesis, National Chiao Tung University, Taiwan, R.O.C., 1992.

[11] Yuan-Sheng Fu, "Hand-Printed Chinese Character Recognition by Multiple Neural Networks," Masters Thesis, National Chiao Tung University, Taiwan, R.O.C., 1993.

[12] Ting-Shan Cheng, "A Neural Network System for Handwritten Chinese Character Recognition," Masters Thesis, National Chiao Tung University, Taiwan, R.O.C., 1992.

[13] Chun-Hsien Chen, "The Study of Neocognitron for Handwritten Characters Recognition," Masters Thesis, National Chiao Tung University, Taiwan, R.O.C., 1991.

[14] Bin Chen, "Recognition of Handwritten Chinese Characters via Short-Line Segments," Masters Thesis, National Chiao Tung University, Taiwan, R.O.C., 1993.